

Release 1.0

88-5

9 May 1988

OBJECTS ON PARADE: NOW FOR SOMETHING COMPLETELY DIFFERENT

Consider the lowly operating system. In general, applications are conceived and built, but mainstream operating systems evolve. There's too much history to start afresh -- as witness OS/2, a follow-on to DOS; the new unified version of the perennial UNIX; IBM's continually evolving suite of systems software. Mature OSES grow weedy and muscle-bound, powerful but hampered by their own vestigial organs and accrued scar tissue. Over the years new features and functions are added, but few are removed for fear of disturbing something that works. Common, widely used functions that may not formally belong in an operating system at all are added so that they will be available to all applications and users. Each piece is inserted carefully to work with the existing mess (and the installed base). Years and years of compromise have produced monsters that work in ways and for reasons no single person can understand. We are now at the mercy of millions of lines of code we can no longer tame or dismantle.

Key Logic, a company founded in 1985 by a group of nine ex-employees of Tymshare, has a new approach: Start over. Begin at the beginning, emulate the essential functionality of existing OSES so that you have some hope of finding customers, and do the job right. People have avoided this approach so far because of the awesome complexity and the hardware dependence of any industrial-strength operating system. But the trick is not to imitate what has gone before but rather to provide its benefits without its drawbacks. An operating system doesn't have to be overly complex, if you break the system down into its constituent parts and let them handle it themselves.

This requires nothing more than the implementation of this season's favorite buzzword: "object-oriented." Object-oriented is in. But what does object-oriented mean as applied to an operating system? The benefits of object-orientation are many, and, in keeping with the granular spirit of objectness, you don't need to be fully object-oriented to gain them. These include the crystallization of complexity into small pieces that can be handled discretely, late binding and information hiding, so that one part of an application need not know the details of another in order to use it. Other benefits include inheritance (for reusability, ==>

CIAO VITTORIO!

<u>INSIDE</u>	
OBJECT OS	1
RISCy software.	
WHO'S ON FIRST?	7
Server solutions.	
Banyan's VINES.	
Oracle's SQL*Mail.	
SOFTWARE RE-ENGINEERING	11
Coining a market.	
A CASE FOR IPSE FOR CASE	13
Atherton's BackPlane.	
COMPUTER LIBRARY	15
Finding ourselves.	
NITTY GRITTY EXPERTS	17
Let the computer play God.	
RELEASE 0.5	18
PictureTel.	
RESOURCES/CALENDAR	19/20

consistency, and avoidance of redundancy), and a closeness of fit with the real world, which contains things far richer than can be represented by flat files or even relational databases.¹ In other words, you can request the results of a service or function without specifying details of how it should be carried out. "List all the people in Alice's department." "Fetch the background on Juan's promotion." "Waitlist me for the flight to Barbados."

With all these benefits, it's a wonder they haven't been applied to anything other than applications and databases until now. In fact, Key Logic has been around for some time: The project began in 1974 at Tymshare, and the company has received advice and funding from board members Gene Amdahl and Tymshare ex-chairman Tom O'Rourke, as well as from H&Q and other VCs. But it may be ready to gain notice now for several reasons: It has hired a new ceo with good industry connections (Marc Butlein from Gartner Group); it has put together one of the most polished marketing pitches we've ever seen for such a new-wave product; it looks close to winning some new commercial customers; and it is working with the DOD's National Computer Security Center with good prospects for Federal certification for type B3 ("Orange Book") security, a high status held by no general-purpose commercial OS. (Ollie North was on a D-rated, unsecured system.) Only one product has A1 status, SCOMP from Honeywell, a communications system in limited use. Multics, also from Honeywell, is rated B2 (below B3).

Finally, in the best welcome-competition tradition, Key Logic may gain credibility from other object/capability-oriented OS efforts now gaining notice -- Carnegie-Mellon's multi-year MACH project, some of which may show up this year on the NeXT machine (but on it only), and Sun/AT&T's Menlo Park Phase 3 project to rewrite UNIX in C++, an object-oriented extension of C, which has just begun and probably won't bear fruit for a couple of years at least.

Object-oriented box

Object-oriented languages such as SmallTalk and C++ (see Release 1.0, 87-3) are flourishing; object-oriented development environments such as Whitewater Group's Actor and Stepstone's (nee Productivity Products International) VICI/Objective C are selling well and broadening their distribution; object-oriented database management systems such as Servio Logic's Gemstone, Ontologic's Vbase (87-8) and Symbolics' Statice (88-4) are reaching commercial status. Xerox spin-off ParcPlace Systems, the vendor of SmallTalk, has just completed its disentanglement from Xerox and is now an independent entity (with funding from Xerox and a couple of VCs). Congratulations, Adele Goldberg!

What could possibly go wrong? Well, the idea could still fail to catch on, because operating systems aren't replaced; they're enhanced. Whatever the merits of a from-scratch approach, customers may be reluctant to try it. And if Key Logic is wildly successful, because its approach makes such good technical sense...then the company is sure to evoke competition (and buy-out offers) from all the people it is now hoping to make its OEMs.

¹Objects don't fully represent reality either, but they come much closer. Being extensible and all-encompassing -- you can put anything inside an object -- they have greater potential to do so.

Objects in context

At the same time that operating systems have been growing more complex, the software community has been developing the notion of object-orientation for applications. Some of these ideas are also turning up in operating systems, notably OS/2 with (eventually) its installable file system (see Release 1.0, 88-3). Operating systems and other systems software (communications, especially) is becoming layered both horizontally and vertically (cf. our discussion of directory services, page 6), so that an application or user can call something without necessarily knowing where it's located or how it works. Dynamic linking of libraries, as in OS/2, allows code modules to be upgraded or extended invisibly. Increasingly, in OS/2 and UNIX and elsewhere, programmers will not program so much as *assemble* application modules into constituent pieces of new applications. All these approaches depend on the features of objects.

Key Logic simply carries the paradigm to its logical conclusion, but instead of treating what used to be applications like OS functions (as everyone is now doing to database services on OS/2 and elsewhere), it treats OS functions as applications. Except for the basic functions of hardware access and memory management, all the system utilities -- schedulers, file managers and other functions -- now act just like applications. (Or *are* they applications? Never mind!) However, unlike traditional applications working in hierarchically-structured environments (and similar to object-oriented modules) they are available as services to each other or to other applications.

Objects in orbit

Key Logic breaks down the distinction between applications and system utilities, and reduces everything to independent functions or sequences of functions. The net effect is to reduce drastically the amount of code in the OS, and to increase drastically its comprehensibility, reliability, and sheer efficiency. Key Logic's generic technology, KeyTECH, comprises nine basic components. Two are "privileged" primary objects, or capabilities, controlling access to the hardware and to system memory by way of access keys stored in the kernel memory. The rest are users of the system with varying levels of privilege, granted by their keys. The basic nonprivileged object is the "domain," which amounts to a virtual machine with its own OS and memory, contained in objects called segments. These are incorporated only by reference and called as needed, so that domains can share resources.

Use of (vs. access to) resources is controlled by "meters" which monitor the use and allocation in a sort of decentralized market system, with prices and kitties (de facto priority assignments) set by a system administrator directly or through algorithms. Again, rather than schedule in a brittle way, the system allocates resources at runtime, for maximum efficiency in a dynamic environment. "Keepers" take care of exception-handling. Domain keepers, for instance, handle emulation of guest OSes.

Objects in performance

Doesn't all this key-switching and message-passing consume resources? Not compared to the traditional method, which overloads the system with static allocations of resources can't be shared. Basically, it's easier to share things on request at runtime than to plan adequately for an uncertain se-

quence of operations. As a batch environment KeyTECH would not be especially efficient, but for today's world of transactions, interactions and exception-handling it's ideal.

Yes, we found this hard to believe too, but we checked with customer John Earle, a consultant at Australia's ANZ Bank, who fairly bubbles over with enthusiasm at the system's reliability (99.9 percent availability) and speed. He cites top rates of 15 transactions per second on a 4341, which is double what he would expect from IMS on that system, and far more than needed by the application, in production since September. And it scales up, with rates up to 400 tps in benchmarks on a 3090. Fundamentally, says Earle, "It never runs out of architecture; it runs out of hardware."

To be sure, it's likely (but hard to test, given the paucity of samples) that KeyTECH's performance has as much to do with its clean implementation as with its architecture. If you were to build a perfectly optimized version of MVS from scratch, it would probably be a lot leaner and quicker than what the world is stuck with now.

Objects on parade

We've all seen those drawings of software with all their layers -- first the hardware, then the OS, then the database and other "foundation" software (such as IBM's OS/2 Extended Edition). The OS is generally an undifferentiated layer of sludge that insulates the software from the hardware, and is so self-referential you can't separate pieces of it out.

Key Logic has crystallized the functions of an OS into discrete objects that are used only as necessary. Right now, its product comes in the form of KeyKOS, a 370-specific system that includes emulations of some MVS functions, such as compilers, access methods and a few environments such as CMS, CICS and X.25; these functions are executed by the segment keepers, exception handlers that trap application requests for these services. To the extent that KeyTECH implementations emulate anything, they emulate OS functions, letting the hardware instruction set show through.

KeyKOS consists of a kernel written in 370 assembler, plus 370-specific functions, plus KeyTECH, a generic set of OS functions designed to be portable to almost any hardware, based on machine-specific kernels which will be written in assembler. The UNIX implementation, KeyNIX, will consist of a similar machine-specific kernel, plus KeyTECH, plus a complete set of functions of POSIX (the /usr/group standard version). While KeyKOS is 370-specific, the KeyNIX will be designed to let C applications be portable from system to system, just like "real" UNIX/POSIX. KeyTECH is designed to slip into an existing environment and look as much like it as possible -- except for speed, reliability, security and efficiency.

In a sense, KeyTECH resembles a RISC chip -- a stark, sleek system that contains the bare minimum, and that can take on many different characters by virtue of the software above it. The result is an OS kernel that takes up only 35,000 lines of code, plus 100,000 lines for OS functions outside the kernel, vs. 3.5 million lines for all of MVS; 600,000 to 700,000 for UNIX; and 1.5 million for OS/2. This is not a truly fair comparison, but it provides perspective.

Objects by design

The Key Logic system sounds so sensible that it feels implausible. How could anything as complex as an OS be made comprehensible? It's a fundamental tenet of dp that systems software is complex and must be managed by highly trained and compensated systems professionals. Of course, it is these professionals who will evaluate KeyTECH/KeyKOS, which may pose a challenge to the company's sales efforts. But the market is changing. Company managements no longer consider DP a black box, unaccountable to the rest of the company. DP must pay its way.

KeyTECH grew out of work done at Tymshare (where it began life as Gnosis, the Greek word for knowledge, and also an acronym for Great New Operating System In the Sky). Tymshare was searching for an efficient OS not to handle time-sharing, which requires separate virtual machines for multiple users and long sessions (cf. UNIX), but to manage much smaller-granularity transaction-processing, which requires efficient creation and termination of processes that exist only long enough for a single transaction (cf. OS/2 and more particularly Sybase's re-rendering thereof). A by-product of the transaction orientation is a notion of security attaching to individual processes and not to whole users, which is a much leakier approach.² And indeed that positioning persists, as Key Logic is going after three markets: security-conscious outfits such as banks and government, because of its superior security features; OEMs, because who else would buy an OS?; and EDI/transaction-processing users, because that's what it was designed for.

Selling to OEMs will be an interesting challenge: Butlein, who managed the large-systems business at Gartner Group, thinks he might be able to crack the UNIX market (now that it is in turmoil, and companies are scrambling for advantage) by giving one or two vendors a way to differentiate their UNIXes -- not in features, but in performance, robustness, and security, always a serious flaw in UNIX. (Can this be done without compromising the current nature of UNIX? Of course not, but many users would welcome the change.) Users are unlikely to adopt KeyTECH wholesale or use it to replace existing systems, but they may use it (as ANZ does) for separate, new applications that are important and performance-sensitive but not fundamental to the rest of their operations -- much as users started adopting Tandem ten years ago.

As operating systems with all their facilities become a greater and greater center of revenue or cost (both per se and for ancillary services and caretaking) they will become subject to the same competitive pressures that affect hardware and applications -- but they will not necessarily be as tied to hardware as they once were. Customers no longer automatically pick them both together. Even as users clamor for standards, vendors want to differentiate around those standards. So perhaps Key Logic has a chance.

²When we log on in today's typical user-oriented environment, any application we run has access to all the files we have access to. Suppose that application is corrupted (yes, the v-word); the virus can now go corrupt the files we have access to. A system such as Sybase's SQL Server, which insulates the server from users and applications also thereby insulates from corrupted user applications -- but not from corrupted processes talking to the server itself. But in a capability-based system, access is much less freely granted and generally attaches to specific processes only.

WHO'S ON FIRST? EXPLOITING NETWORKS WITH DATABASES

In order for heterogeneous networks to work together effectively, they need to know about each other. Most networks have a file of users and resources, but you (or the application) need to know the right address to get to them. There is no general standard, and so each user/application needs to know the naming conventions across the networks it is using. Gateways provide protocol translation and access, but don't help people operate within other environments. In other words, to talk to someone in France outside of Paris, French isn't enough; you need to know a little geography.

The two articles following address two different aspects of this issue: system underpinnings, in particular directory services, and applications, in particular database-based E-mail. Both of these, of course, are ideal support for and examples of potential OS/2 server applications, although the specific items here are (initially) UNIX-based.

BANYAN: A NET OF PRESENT VALUE

Directory services that know where to find things are necessary for effective network management. To optimize database queries, for example, it helps to know details as picayune as the number of records in the tables a query joins; to optimize communications paths, it helps to know the relative costs and loads of each possible path, both in general and in real-time.

As people start to connect up more and more widely, the greater the necessity for some common underpinning, a cross-system standard, that could know all about the resources and people hooked up everywhere, and how to get to them. Unfortunately, most network systems are hermetically sealed: If you buy everything from a single vendor, you have some hope of a coherent system, but otherwise each network regards the next one as foreign territory. As soon as the going gets tough, the facilities give out, and users must rely on patchwork methods to communicate outside a narrow boundary. All the network protocol standards -- TCP/IP, SNA, NetBIOS, etc. -- don't win you anything in this regard either: They help send messages, but they don't worry about finding the right party at the other end.

Much of the world is waiting for X.500, a forthcoming standard that will obviate these problems once everyone adopts it. (It's a follow-on that will provide support for the X.400 E-mail standard among other things.) In the meantime (which could last years), one of the more advanced network vendors in this regard is Banyan Systems, long considered a laggard behind Novell and 3Com in the pc network world. In fact, while 3Com and Novell were learning about retailing and marketing, Banyan was busily serving a high-end market with capabilities that are now appealing to a broader universe. Unlike 3Com and Novell, which defined their basic charter as hooking pcs up to a network, Banyan was started in 1983 by a bunch of guys from the mini business (Data General, mostly). Rather than work on extending pc applications to communicate across a network (or to retrieve data from a remote file server), Banyan was concerned with supporting multi-user applications as server applications, ones that acknowledged the presence of an intelligent pc rather than a dumb terminal at the other ends of all those wires.

From the start Banyan has used a UNIX subset as its host operating system for VINES (VIRtual Network System). 3Com and Novell favored proprietary systems, and are now adopting OS/2 as a server standard (Novell only partially). Banyan's decision looks pretty good as UNIX gains momentum, but Banyan is leaving a lot behind unless/until it adopts OS/2 as well. For now, Banyan's VINES supports OS/2 only as a client OS; and correspondingly, VINES supports UNIX not on client stations but only on servers.

Banyan's roots

Aside from these standards issues, Banyan's VINES offers impressive functionality, such as wide-area connectivity, hardware independence and ease of use, especially in multi-server environments. This has gained it a widespread presence on Wall Street and in government, with 150,000 nodes installed.

Most interesting -- worth licensing and standardizing -- is its directory or name service, StreetTalk[™]. StreetTalk underlies much of VINES' ease of use and its ability to provide transparent access to other VINES servers and their resources, even across a dial-up line or other wide-area network. Basically, StreetTalk is a distributed, partially replicated database with an interface for applications. Its software keeps it up to date and consistent; when a new server is installed, it immediately communicates with other servers on the network so that everyone knows about everyone; likewise, each server keeps track of the resources it controls and updates other servers. The system handles the management of multiple resources (three fungible printers, say) which lets users operate without knowledge of where people or resources are located. All you need to know is the file name or the service you need, not the path to it. Name duplications are resolved by notifying the system administrator when the second name is added.

People installing applications based on VINES API need not know the intricacies of a system set-up, and services can be moved without reconfiguring each of the applications that rely on it. Like any directory service, StreetTalk provides a real-time logical interface that people and applications can talk to, and collects and maintains all the system information.

But for now (and until the arrival of X.500) Banyan seems to have an edge: Aside from Banyan, Locus and Apollo have the strongest underpinnings for their distributed Oses, but both are in the business of selling machines or Oses rather than networks to support other people's installations. Similar services from 3Com in 3+Open are centralized (or distributed without provision for automatic propagation of updates), and therefore less fault-tolerant and harder to implement on a multi-server network. Novell and IBM's pc LAN systems do not have a directory service, in the sense discussed here. IBM has a name facility in PROFS that doesn't extend outside that environment, and is working on something more general. Sun's Yellow Pages are fully replicated, which uses lots of resources, and the system has automatic but not immediate updating capabilities. DEC has announced a system that looks promising but isn't out yet. Finally, Microsoft's LAN Manager (built with 3Com) will operate without a full standard directory service and will do subsets of the evolving X.500 specifications until a complete X.500 standard is promulgated. Banyan, of course, will also support X.500 within VINES with whatever changes are necessary to maintain transparency to existing users.

As you might expect, StreetTalk's powers end at VINES' boundaries (although far-flung VINES systems can be linked transparently through remote connections), but Banyan is publishing its interfaces in the hope that other software developers will use them so that their users will get the same transparency users get from the VINES services (mail, gateways, etc.). Second, Banyan will consider licensing the service itself so that Banyan and other network vendors can provide transparent access between VINES and other networks -- rather than making users stop at gateway boundaries as they do now.

To date, Banyan has been extremely successful within its own installed base, but has failed to market aggressively. That situation is changing with the recent acquisition of president Dick Meise from Convergent Technologies, and vp marketing Peter Simon from Lotus. Another sign is Banyan's recent OEM deal with Wang, which should help the company extend its reach in marketing even as it helps its customers extend their reach technically. The company is proud of StreetTalk but it wants to husband its resources carefully. Says director of product planning Dave Williams, "We want to parlay this not just into money but into strategic alliances."

ORACLE'S SQL*MAIL: SMART E-MAIL

Directory services are designed to foster communication within a system among applications and resources, and to provide facilities for transparency to applications such as E-mail. By contrast, E-mail is an application whose primary purpose is to foster communications among people. In both cases, these services are enhanced with the use of an underlying database, which ensures consistency, integrity, efficiency, and all the other virtues databases bring to applications in general. If you put a database underneath your E-mail system, for example, you'll end up with something far more powerful and extensible than the systems common today. It won't just hand you your mail; it will sort it and even answer some of it for you.

So what do E-mail and databases have in common? They turn into each other if you squint. A point-to-point E-mail system is file transfer; a server-based E-mail system (however it looks logically) is just a set of mostly transient files with some application code. At the moment, most E-mail systems are like the applications of yore: the data and the application specifics are inextricably intertwined. But consider the advantages of borrowing from the powerful database underpinnings already in existence to strengthen an E-mail system -- integrity (you can't delete a message that hasn't been properly disposed of, however you define "properly"), storage and retrieval by classification, the use of rules and triggers to handle certain kinds of messages automatically, calculations and comparison, standard forms with defaults to make messages easier to compose and interpret, and, if you have a portable database, inherited portability. If someone has already gone to the trouble of building such a database, why not share in the fruits of that effort by putting E-mail on top of it?

With database-based E-mail, conditions can trigger actions: When the date value is April 15, print the message, "You're overdue!" Or if the count of unanswered messages from the same person is greater than three, change the priority to "urgent." Filtering mail or acting upon it is an application. For example, you might have a program that assigns priority to messages

based on combinations of values -- the caller, the subject (was it one of several with assigned weightings, such as the Monday meeting agenda, "the problem," the space project?). All this can be made easier for sender and receiver with message templates -- forms -- with default values.

Aside from typical database functions, E-mail requires little more than a notification service; rather than wait patiently for a user to make a query, the system notifies a user in realtime or at specified intervals when new mail has arrived (or when old mail passes some response deadline). It also needs special fields for free text, and pointers for the inclusion of files with the messages. A mail message is a transaction, no more.

As a group uses an E-mail system, it will build up a corpus of standard messages, gradually reducing the exceptions to a small proportion of the total. In this way, E-mail can fulfill the goal of much automation: handling the routine stuff so that people can devote their time and energy to the non-routine. As in any database, the advantages are many, including the avoidance of redundancy -- a single message can be sent to hundreds of people; a standard message can be personalized for hundreds of people, with only the varying fields (not the common ones nor the calculated ones) needing storage. It's like real-time mail-merge: Your message is created when you ask for it, just as a database view is created when you make a query.

Information Lens reincarnated

Our favorite E-mail system is not commercial; it's the Information Lens project at MIT, led by Tom Malone. Built on Xerox LISP machines with a neat combination of AI, database management and simple good design, I-Lens acts as an intermediary among users, helping them to filter and respond to their messages and organize the resulting files. Ever since we wrote about I-Lens in September 1986, we've been waiting to hear of a commercial expression of this idea. After all, isn't anything that started out on Xerox D-machines guaranteed to be fashionable nowadays?

At last we're about to see a first pass at one, with the power of an underlying database management system but without (yet) Information Lens' flexible AI facilities for automating users' handling of their messages (some of which, to be sure, can be accomplished with procedural programming just as effectively). Oracle Corp. recently hired a member of the Information Lens team, and is about to announce a database-based E-mail system which embodies some of its concepts, and which carries the hooks to do much more. (Although the I-Lens alumnus is not working directly on the project, he has inspired Oracle's thinking.)

Oracle's SQL*Mail, due to ship later this year, lacks the full smarts of Information Lens, but it packs a lot of power and price-performance. The system's appeal is two-fold: its intrinsic worth as an E-mail manager (not just a delivery vehicle), and its inherited ability to run across a variety of hardware platforms, based on the ability of the Oracle dbms to do the same. For sites with an Oracle dbms already installed, it will be a relatively simple and cheap application add-on.

Each Oracle E-mail message is a database record, with a number of pre-defined fields, such as To, From, Date and the message-text field, plus internal invisible ones (postmarks, etc.). Users can also define an arbi-

trary number of additional fields. The records created are subject to the same powerful manipulation capabilities as any Oracle database records: They can be sorted, queried, viewed, filtered, and otherwise acted upon.

As in Information Lens, one can create pre-configured, partially filled-in message types (records with default values) to cover standard events (trendy people might call them forms): meeting notifications, customer contact reports (for requests for information, orders, complaints, etc.). In fact, the E-mail system will ultimately work as a front-end for a regular database-based application using the Oracle SQL*Mail API, as well as a means for people to communicate with each other. An order message, for example, is the same as an order entry, but it gets routed to the credit manager first.

This approach makes it easier to handle both highly structured messages and less structured, traditional E-mail. Oracle's SQL*Mail should provide an easy transition -- and it also provides easy exception-handling, as forms with non-standard information can be routed by the user to the person who can do something about them. (In the future, the system will do the routing automatically, as in FCMS's Staffware, described in Release 1.0, 87-12.) The user gets the added benefit of a subject/content-oriented correspondence database, instead of the traditional chronological catalogue of mail (perhaps enhanced with a free-text search capability that SQL*Mail at first will lack). In the long run, mail systems will grow even richer, with object-oriented underpinnings that will allow users to manipulate concepts as well as structured fields, and natural-language parsing that will let a system interpret and handle messages that don't follow standard formats (cf. Release 1.0, 88-2). That's the ultimate goal of thing Information Lens will evolve towards even as Oracle works for commercial acceptance of its first-pass commercial implementation.

SQL*Mail will be available first on SUN UNIX and VAX VMS, and, points out Oracle sales chief Pete Tierney, for the real high-volume stuff you can use Oracle's forthcoming transaction-processing facilities. Be that as it may, SQL is now a language for looking at your mail as well as at your data. But as in any other use of SQL, it is best hidden from the user, behind a bevy of standard forms, canned applications, and other user-friendly clothing, as in SQL*Mail.

SOFTWARE RE-ENGINEERING: DEFINING A MARKET

How do markets get created? We recently had lunch with a friend who's trying to create reverse engineering much as James Martin created 4GLs and CASE. It's an interesting process to watch -- and join.

Rich Currier runs a company called Panoramic Inc. (see Release 1.0, 86-5), a vendor of "software glue" that front-ends and links a variety of heterogeneous mainframes, minis and other systems. With a little programming, it can combine interactions with each system's applications and present them all to users via a single screen (per user) as a single, seamless application. Rather than invade the complex, fragile systems it front-ends, it looks to them like a user, enabling system builders to graft new stuff onto old stuff with minimal disruption, like adding a motor to a sailboat without dismantling the hull. The formal term for this is "session management." In essence, it's a form of what we call retrofit technology: Rather than rebuild existing systems, it reconfigures them into something more useful without reworking their guts.

Software re-engineering takes up where retrofit technology leaves off. A little more radical, re-engineering typically involves taking something apart to put it back together again more effectively, perhaps using newer technology (hardware, software or architecture) or with greater attention to user-friendliness. Software re-engineering is about to become a big market: CASE conferences regularly have re-engineering tracks, and customers are eager to learn how to convert the mountains of old code that are strangling them into new, effective systems. The installed base of close to 100 billion lines of mostly COBOL code is too old, inefficient and unfriendly to keep in its current state, but too big and valuable to squander. It has to be re-engineered.

As a long-time participant in the mainframe software business, where he worked for Software AG and Walker Interactive and saw the rise of 4GLs and databases, Currier wants to be in on the start of the software re-engineering revolution.

The market works!

The first step: Run a conference. Currier has signed up to lead two seminars on re-engineering for Digital Consulting Inc., a savvy seminar and consulting outfit that has made lots of money by introducing dp managers to concepts such as 4GLs, CASE, etc. The two conferences will take place next fall (see calendar, page 20), with the usual assortment of company pitches, luminaries, and Currier himself as chief luminary.

For DCI the trick is to get vendors to pay to attend, and customers to pay to listen. But it's an easy trick: The conference itself is mouse nuts. For the vendors, the appeal is not just reaching the people who attend, but the impressions created by the brochures DCI mails out -- your company name and a descriptive paragraph in front of thousands and thousands of qualified buyers who will become familiar with the companies and the buzzwords therein. For the users, it's an easy way to get the benefit of a couple of days of sales calls all in one chunk.

Next step: Get your friends in the press to write articles...

As it happens, we've written about the topic many times already (see Release 1.0, 86-4, 86-5, 86-12, 87-6, 87-10, 88-4) and we believe it deserves all the mention and recognition it can get. Like many markets, software re-engineering is really just a state of mind. Just as multi-user systems all became distributed after Datapoint coined the term in the late 70s, or clever algorithms become AI at the behest of some marketing type, so can many activities be considered re-engineering if you look at them right. Anything that helps you understand existing systems qualifies, as does anything that *attaches* to existing systems. Anything that helps build new systems qualifies, if it pays some homage to old systems.

Maintenance qualifies. Indeed, the first stirrings of the re-engineering movement came with the efforts of Nicholas Zvegintzov, editor of the cleverly named ^{Maintenance} SOFTWARE \ NEWS.

Zvegintzov's movement is kind of a feel-good society for downtrodden maintenance programmers who don't get no respect. The problem with these movements is the same as the problem of unionizing waiters: No one is really a waiter; they're all actors or novelists waiting for a big break, and they don't identify with the waiter cause. Likewise, maintenance programmers, in this upwardly mobile society, are all on the way to being analysts, designers or managers.

Advertisers in ^{Maintenance} SOFTWARE \ NEWS include KPMG Peat Marwick, Language Technology, ViaSoft and a host of other companies, all of them vendors of tools to examine or rebuild systems, not just to maintain them. The importance of a more glamorous positioning to these companies is obvious: It may help them sell more software. Likewise, it will help Rich Currier and DCI make money, and keep the economy whirling.

But the importance of recognizing and recovering the value in existing software extends beyond this industry to the productivity and efficiency of the country as a whole. Yes, much of what the programming community has built deserves to be thrown away, but there's a lot of value there if we could only refine it or, to use the term of Language Technology's Eric Bush, reincarnate it.

A new way of looking at things doesn't necessarily just sell products: It makes people experiment in new, possibly fruitful ways. It makes us ask some appropriate questions: What do we have already that's of value? How can we take what we have and reuse it? How can we take what we have and improve it? Rather than sit around and wait for the ultimate answer, let's take what we've got and fix it. Let's make existing systems more maintainable and comprehensible both to programmers and to users. Just as waiting for the best is the enemy of better, so is dreaming of the new the enemy of retrofit. Retrofit is clearly better than what we have, while new is a long way off, maybe never.

(How does Key Logic, with its from-scratch new OS (page 1), fit in? First, the rest of the world should be doing something while they wait for KeyTECH. Second, Key Logic is doing retrofit technology to the extent that it is providing emulations of existing OSes so that existing applications need not be redone to run on KeyTECH; what's new is the OS technology beneath those applications. And third, KeyTECH in its native mode, which will look object-oriented to developers and programs, will likely be used primarily for new applications, not re-engineered ones. Glad you asked.)

A CASE FOR STANDARDS FOR CASE

Last issue we talked about the importance of a repository for CASE, and the standards that the ANSI IRDS (Information Resource Depository System) committee is working on. However, that standard has more to do with the representation of program elements than with how they're managed, and is targeted towards the commercial world. Atherton Technology (which unfortunately had to give up the name of CodeSmith for legal reasons) is promoting a standard to manage the process of development and the integration of the tools (not their data).

Atherton Technology was founded in 1986 by Greg Lynott and Bill Paseman, both from computer-aided logic design vendor Daisy Systems, to build support tools for the system development process. They were joined by Daisy founder Ram Banin, now president and ceo, and have since raised \$7 million (soon \$11 million) in funding from Kleiner Perkins, Mayfield, Menlo Ventures and Morgan Stanley (see Release 1.0, 87-6). The company's vision reflects its managers, and is somewhat semiconductor-oriented: In brief, Banin feels that system builders tend to be prima donnas who would rather do everything themselves, whereas hardware designers are perfectly happy to wire together pre-configured ICs, adding pieces of custom logic here and there as needed.

One ultimate goal of the company is to be a supplier of such software ICs (an idea actually implemented by Stepstone, nee Productivity Products International, with its class libraries of software objects), as well as the tools to reuse and combine them. But it turned out the market has more need for tool-integration than for tools: It needs a common substrate into which to hook these software ICs and tools, and a system for managing them, a so-called IPSE, for Integrated Programming Support Environment. Moreover, these software objects are far more malleable and subject to infinitely greater variation than ICs, and can't be comfortably catalogued in a book like ICs. Fortunately, as software, they're susceptible to electronic storage and manipulation in a way that ICs are not.³

Thus was born the Software BackPlane, a system for managing the code elements created by other tools and represented in structures the BackPlane can ignore. In its basic mode, the BackPlane treats the systems elements as black objects, not peeking inside them but carefully monitoring their provenance and status. The system maintains links between specs, diagrams, code, data definitions, documentation and other effluvia of the systems development process, and among the counterpart versions of each (the possible permutations are awesome); it can assemble collections of versions into reproducible releases. The system also manages the administrative flow that gets the work done, controlling the work/rework/approval/integration cycle in successive iterations as larger and larger segments of a system are constructed, checked, and integrated into a whole. These and other procedures can be added by users as processes designed to trigger whenever elements are checked in by a user. The BackPlane also includes Project Soft-

³...although that is changing as custom-logic design systems such as Daisy's and Silicon Compilers' and erasable, programmable ROMs gain sway. In fact, the distinctions between hardware and software are lessening even as Atherton pushes hardwards in the software business, Silicon Compilers et al. push softwards in the hardware business, and RISC makers leave much of the burden to compiler builders.

Board, an E-mail system integrated with project-control software that lets people communicate free-form but also supports more structured interaction about the status of the work in progress.

The system also provides facilities for integration with tools, making the appropriate data available to the appropriate tools, and restricting access as specified. It can assist a closely integrated tool in working with individual fine-grained objects within a file.

Contrast and compare

Although the distinction is more one of culture and approach than technical capabilities, Atherton is targeting the aerospace/engineering market rather than the MIS/COBOL commercial crowd targeted by IRDS. The biggest technical difference is that the BackPlane provides no support for data formats or exchange, for moving, say, specs to code generators. The BackPlane expects the data transformations to happen within the head of the engineer using the system, and simply supports him with tools that support Ada or C development -- generally, language-specific editors and simulators and debuggers and cross-compilers, as well as design/documentation tools from third-party supporters such as IDE and Nastec.

Unlike IRDS, the Software BackPlane is a networked system that performs these processes, rather than a specification for data representation and exchange that can be implemented on a variety of database platforms. The BackPlane manages the development process, not the elements developed, with rules for managing program elements, while IRDS is a standard for the representation of individual elements but does little for overall coordination and control. That's a job for an application that could be built around the repository -- in fact, an application such as the BackPlane. Atherton certainly hopes to support a data representation standard when one solidifies.

Rather than stuff all the system elements into a single straightforward dbms such as DB2 or a clone (the IRDS approach), the BackPlane leaves them where they are, under the control of a variety of tools. The BackPlane acts as an object-oriented integrator that knows where things are physically and can assemble them as necessary to give each user or application the appropriate subset, appropriately presented.

The BackPlane is designed to work on a variety of hardware platforms with a minimum of porting effort (on top of its own OS interface). It is available now on the Sun 3 and on VAX with VMS, with DEC Ultrix and other UNIX implementations to follow. The company has scored a coup by winning a recent deal with DEC, which still lacks a CASE support environment such as Apollo's DSEE or Sun's NSE, let alone all the functions of the BackPlane. The two companies will work together to port the BackPlane onto DEC's Ultrix (UNIX) environment, and to integrate it with DEC's VAXset tools.

The significance of Atherton's effort lies in its attempt to standardize its facilities. Even as CASE tools solve some development problems, they create new problems of incompatibility among themselves. Building islands, however desirable they might be, creates an intractable need for bridges. Atherton is trying to provide some landfill.

A TALE OF TWO VENTURES: ZIFF'S COMPUTER LIBRARY (AND CID)

By way of full disclosure... It's rare to write about something to which we have so many connections: First, we used to work at Ziff-Davis. The particular effort we worked on there was an electronic newspaper, Computer Industry Daily, to be delivered on paper or (ultimately) online, not an archive of abstracts, but you can see the correlations. Second, we're a big fan of text-based applications/services. And finally, we would be a typical customer of this product, and we'd also like to be a supplier, subject to Ts and Cs -- perhaps a three-month time lag? (Product manager Seth Alpert scrupulously avoided discussing this possibility before publication of this commentary, or did he just not take us seriously?)

Ziff Communications, publisher of PC Week, PC Magazine and a host of other trade and user publications, is launching Computer Library this week (for shipment this summer). Computer Library is a monthly CD ROM disk compendium of the full text of 10 computer publications including the Ziff stable plus, of course, Lotus Magazine, Communications of the ACM, and Microsoft Systems Journal, whose publisher Jon Lazarus used to work at Ziff-Davis and co-founded CID. Abstracts of 110 other trade and business publications are also provided on the disk. Each disk extends back 12 months, and arrives about three to eight weeks after the corresponding print publications. The pricing is aggressive: \$720 per year, plus (if you want it) \$635 for a Sony CD ROM drive. The text-search software is a runtime version of BlueFish, in its first emergence since the acquisition of BlueFish vendor Computer Access by Lotus. Lotus gets a license fee, as do the suppliers of the full-text publications. (Abstracts don't breach copyright protections.)

But Ziff provides value-added beyond distribution: It is using a cadre of 25 abstracters to catalogue, summarize and index the material. Development vp Seth Alpert estimates the start-up cost at \$1 million for anyone who might attempt to compete with Ziff, which already has an edge over everyone except Pat McGovern's International Data Group with its array of computer publications. (Ziff lacks a mainframe book to match IDG's Computerworld, but hopes to get full-text rights to Datamation, now owned by Cahners.)

Instant analysis vs. archived facts

Why does this venture make sense (where Computer Industry Daily didn't)?
Let us count the ways:

- It requires minimal editorial judgement/expense, beyond the low-level data entry (by tape, scanner, or yes, retyping) and abstracting, a process Ziff already knows from its on-line Information Access abstracts database business. The basic editorial material will come from third-party sources who have already established themselves. Management problems will be correspondingly minimal, whereas CID was trying to build a newsroom of analysts who could write on a one-day turnaround -- a fundamental contradiction, we now believe.
- It's priced reasonably, no more than a piece of good software (vs. CID's list of \$1500).
- Distribution costs are minimal, with delivery twelve times a year instead of five times a week.

- The fundamental product concept makes sense. The Computer Library venture seems to be coherent, manageable, and genuinely different from anything else available, unlike our own patched-together, politics-riven scheme to take on the big guys on their own turf -- daily news coverage. CID was founded on the shaky premise that people need information overnight. We discovered that if it was important it showed up in the Journal, the Times or the Merc (or on the grapevine, if it concerned your competition); otherwise, it could wait till the weeklies came out. Worse, one enthusiastic beta reader (not paid, of course!) told us that he loved it; he saved all his copies each week and took them home to read over the weekend!

By contrast, Computer Library provides not just the information but a way to find it (and a convenient way to store it; we have about two CD ROMs' worth of information on the floor in our office, but most of it is inaccessible -- and a safety hazard!). As a potential user, we don't have trouble finding out breaking news; we have trouble finding something we read or clipped five months ago -- or something we ignored then but we wish we had now. As we did the demo in Alpert's office (after satisfying our curiosity about our own name), we found some useful information about the other companies in this newsletter -- information we wish we'd had the week before.

Where CID addressed people who need to know stuff right away (mostly professional journalists, analysts, and other third parties, but few of the ceos we hoped to reach), Computer Library addresses a much larger market of computer users, as well as marketing and sales types who need to track customers and competitors -- in fact, virtually the entire 150,000 subscriber base of PC Week, which will be the first target of Ziff's telemarketing effort. (Ziff has found that 80 percent of its readers keep at least six months of back issues.)

These people can use it to find products they need ("modem" in the same paragraph as PS/2 or "Micro Channel," for a new-style modem), product reviews, installation and usage tips, and the like. People like us could use it to keep our office clean, knowing that we have access to all that information without seeing it on our floor every day. And with the easy retrieval, we might even use it!

From the publishers' point of view, Computer Library provides broader exposure, although Ziff still hasn't addressed the problem of ads or cannibalization of existing readers. The royalties are not large, we suspect, but the deals Ziff is cutting don't require exclusivity: in other words, what's to lose? We think the media are different enough that cannibalization will be a minor problem, but advertisers at some point are going to want a way in. In the meantime, there are plans for providing the Buyers' Guides (a PC Week staple) in 1-2-3 format.

We're delighted to see a practical use of CD ROM, and at a price that should expand the market. It may seem a little self-serving to provide a database of information about the information industry to the information industry and its customers, but that's obviously where the early adopters are. Seth, we're ready to negotiate!

NITTY GRITTY EXPERTS: LET THE COMPUTER PLAY GOD...

Somehow we let ourselves be inveigled into meeting with this game company -- something about expert systems, human interaction, no video. Maybe it really would be different. In the conference room the two people had laid out a tableful of pamphlets, a garish game board...

A computer's intelligence is key to the progress of "Star Saga: One, Beyond the boundary" (from Masterplay Publishing), but it acts as many people imagine God does, guiding things gently without much physical presence. Star Saga is a groupware game: Its players are guided and offered options by the "Computer Game Master," but their primary interaction is with the other players and with the (fantasy) world around them, represented by the playing board and by texts the computer directs them to read. The computer directs the progress of up to six players, and can adjust to their presence or absence at any time -- a handy capability for a game that often takes 60 hours. Each player is a character with personality and motivations but a non-gender-specific name (impossible in most languages save English); the goal of the game is not to score points per se (nor beat the other players) but to fulfill one's own mission in life. (Of course, some people do consider beating others their mission in life.)

Three of the game's authors, including Masterplay co-founder Mike Massimila, played Dungeons & Dragons together at Harvard; co-author and Masterplay president Andrew Greenberg is the author of the Wizardry series published by Sir-Tech. Their vision of the universe -- benevolent, somehow religious, and fervent in support of cooperation, not competition -- pervades Star Saga and informs the rules that the Computer Game Master executes.

No accident that CGM has three letters, of course. Using a model of the game world that is continually updated as players report their actions (location, resources, skills, situations), CGM offers them choices, presents them with information by sending them to the proper readings in the text, and generally takes part in determining their fates. CGM's expert system is programmed to assist those far behind in attaining their goals, and to stall those too far ahead. (Greenberg stresses that this also provides for a better playing experience, as no one gets too discouraged or walks away with the game.) Each player works his way across a board of multi-colored planets with a variety of environments and challenges.

This is not the kind of expert system tool we're used to encountering, but an incredibly rich tapestry of imagined locations, people, situations, challenges, moral dilemmas, foreign life forms, more texture than structure.

But what makes it a fascinating sociological/psychological artifact as well as a game is the CGM. Unlike many people's concept of God, it interferes actively, if still inscrutably, with people's daily lives. Masterplay's God is clearly benign. We're looking forward to seeing the visions of CMG in the knock-offs that Star Saga will inevitably evoke. (Nothing this rich could be cloned.) Could the programming of a Computer Game Master be the ultimate character test?

RELEASE 0.5: PICTURETEL

Personally, we've always been leary of video-conferencing: It's harder to multi-task on the phone (open your mail while talking, say) if someone is watching. But for conversations worth devoting one's entire attention to, video-conferencing is desirable. It provides a broadness of communication bandwidth that almost replaces the 200-pounds-per-several-hours achievable by airplane -- but instantaneously. In fact, video-conferencing is inevitable; it's just that the road to adoption is a little bumpy. Things tend to make conceptual sense long before they make practical sense. The feelings of discomfort have more to do with newness than with technology, the system itself - as any teenager who lives on the telephone can tell you.

Perhaps no other year has lasted longer than the year of the network, with the sole exception of the year of the video telephone. After years and years of just-round-the-corner status, video conferencing may be about to take off. It's at the very bottom of the slope fax is now careening upwards, following last year's personal copiers and car phones, and the decade's personal computers.

Video telephones are still at the very high end of the market; a typical unit is still so big it couldn't even fit in a Ferrari Testarossa. But the usual dynamics are now coming into play: PictureTel, one of the two key players in the market, has just announced a new release of its video conferencing system with double the performance and half the cost. A starter set for a single site (you still need two to communicate) now goes for about \$30,000. We conducted the second half of our interview over the system using switched 56-kilobit lines (now available in 85 cities). Vp marketing Randy Smith, in living color, started looking smeared and jumpy whenever he moved too much, but it was a quality you could get used to (remember TV in the 50s?). PictureTel uses software algorithms to compress the data -- transmitting the differences between screens rather than whole screens each time, and converting motion into vectors. The performance improvement was achieved by sending a new software cartridge to the company's 100-odd customer sites. (The cost was reduced because of a new coder-decoder.)

And there are external factors too. Travel is becoming more unpleasant and costly. With the nationwide shift to an information economy, just-in-time and turnaround time are concepts that apply to the activities of executives and managers as well as to manufacturing workers. Video conferencing can help speed office "production" as well as save money. In fact, beta users are far more conscious of the time shaved off projects and decision-making than of the direct cost-savings. In our speeded-up world, this counts!

Despite its name, PictureTel has no formal connection to AT&T other than a co-marketing agreement. It was founded in August 1984 by two 23-year-old MIT-trained engineers, who are still there, and funded by a \$4 million public offering in 1984.

RESOURCES & PHONE NUMBERS

Ram Banin, Greg Lynott, Alan Codkind, Atherton Technology, (408) 734-9822
Dick Meise, Dave Williams, Banyan Systems, (617) 898-1000
Ann Hardy, Marc Butlein, Key Logic, (408) 496-1090
Andrew Greenberg, Masterplay Publishing, (813) 888-7773
Pete Tierney, Charlene Kabcenell, Oracle, (415) 598-8157 or 598-3566
Rich Currier, Panoramic, (408) 973-9700
Randall Smith, PictureTel, (617) 535-7700
Nicholas Zvegintzov, Software (Maintenance) News, (718) 816-9036
Seth Alpert, Ziff Communications, (212) 503-3523

COMING SOON...

- Database styles: Oracle, Sybase, Interbase, Cullinet, IBM et al.
- More groupware.
- Parallel processing.
- Software evolution.
- Viruses.
- Nitty-gritty airline experts.
- And much more...

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers the pc, software, CASE, groupware, text management and connectivity markets, and artificial intelligence. Editor & publisher: Esther Dyson; associate publisher: Lisa Braden; circulation manager: Hyacinth Frederick; secretary: Anita Fowler; copy chief & consulting editor: William M. Kutik; guest editor: Mark Seiden. Copyright 1988, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we gladly arrange for reprints or bulk purchases. Subscriptions cost \$395 per year, \$475 overseas.

RELEASE 1.0 CALENDAR

- May 15-18** **Software Maintenance Association annual meeting and conference - Chicago.** Feeling good about software maintenance.... With Robert Glass and Gerald Weinberg. Contact: Judith Marx Golub, (415) 852-1640. (See page 11.)
- May 15-19** **Human factors in computing systems - Washington, DC.** Sponsored by ACM groups and the Human Factors Society. Contact: Sylvia Sheppard, (301) 369-2422, Scott Robertson, (201) 932-2911, or ACM (212) 869-7440.
- May 15-20** **International Communications Association 41st annual conference - Anaheim.** Sponsored by the International Communications Association, with speeches by Peter Keen and Bernard Flanagan, vp marketing of The Wall Street Journal. Contact: Lee Flowers, (214) 233-3889.
- May 16-18** **Hammer Forum 88 - White Plains, NY.** "Electronic marketing: information technology in sales and marketing." With talks by David Liddle (Metaphor), Fred Luconi (Apex), Malcolm McKinnon (Prudential), Gian Fulgoni (Information Resources Inc.), John Thompson (Index Group), Stephen Cone (American Express). Sponsored by Hammer Information Systems; moderated by Michael Hammer. Call Pam Davis, (617) 354-5555.
- May 17** **Advanced trader workstations - New York City.** Sponsored by Trading Systems Technology; moderated by editor Corey Bock. With panels of vendors and users, including Russell Hovanec of Shearson and Ian Hobson of Hambrecht & Quist. Contact: Kimberly Arrington, (607) 772-8086.
- May 17-19** **Amy Wohl's dialogue - Boston.** With Pete Peterson, WordPerfect; Henry Ancona, DEC; Tony Kobine, FCMC. Sponsored by Wohl Associates. Call May Giacalone, (215) 667-4842.
- May 19** **Telxon at the analysts' - New York City.** Sponsored by the New York Society of Security Analysts. Contact: Lourdes Figueroa, (212) 344-8450.
- May 23-27** **Folio:Show - New York City.** Sponsored by Folio: and Publish! magazines, with conferences, user group meetings, exhibits for professional users of desktop publishing. Call Pamela Dimond at (203) 358-9900.
- May 25** **Esther Dyson at the BCS - Boston.** Sponsored by the Boston Computer Society. "On beyond numbers..." Where the next generation of applications will lead us. Contact: Jonathan Rotenberg, (617) 367-8080.
- May 30-June 3** **Eighth annual workshop on expert systems and their applications - Avignon, France.** Sponsored by a number of societies, with special-interest days on defense, maintenance, and medicine. Keynotes by Herb Schorr, IBM; and M.-O. Cordier, U of Paris. Contact: J.-C. Rault, EC2, (331) 47 80 70 00; Fax: (331) 47 80 66 29.

-
- June 1-3** **CASE Benchmarks** - Dallas. Rather than just present companies and tools, moderator Vaughan Merlyn controls the proceedings and *compares* the various tools on a common scale. Sponsored by Digital Consulting. Call Scott Dorman, (617) 470-3870.
- June 1-3** **SIGMOD** - Chicago. Sponsored by ACM-SIGMOD, the special interest group on the management of data. Contact: Dina Bitton, (312) 996-0142.
- June 6-8** **Expert communication 88** - San Jose. Sponsored by Graphic Communications Association. Applying AI to design, content, process, etc. Call Marion Elledge, (703) 841-8160.
- June 6-8** **Enterprise networking event** - Baltimore. Sponsored by the Corporation for Open Systems and the MAP/TOP Users Group. A festival in support of motherhood and apple pie. Call Mark Cocroft at GM Public Relations, (313) 986-5719.
- June 7-9** **Executive information systems conference** - Houston. Sponsored by Office Sciences International. Contact: Joel Levy, (201) 750-0085.
- June 13-15** **PC Tech Journal systems forum** - San Francisco. Sponsored by PC Tech Journal. For people who consider the pc a tool, not a 1-2-3 machine. Contact: Beth Hornung, (800) 544-PCTJ or (301) 740-8300.
- June 13-15** **Conference on R&D in information retrieval** - Grenoble, France. Co-chaired by Gerard Salton; sponsored by ACM. Contact: Gerard Salton, (607) 255-4117.
- June 15-16** **Workgroup computing: current status and outlook** - New York City. Sponsored by Brandon Consulting Group. Groupware in all its varieties; a close look, with the usual group of speakers. Call Frank Lesh, (212) 977-4400.
- June 19-22** **Congress VI** - Paris. The world computing services industry gets together. Sponsored by national trade organizations, including our own Adapso. Contact: Phyllis Cockerham, (703) 522-5055, or Diana Kirby, London, (441) 405-2171.
- June 21-23** **PC Expo** - New York City. End-users en masse. Contact Steve Gross, (201) 569-8542 or (800) 922-0324.
- July 12-15** **CASE '88** - Cambridge, MA. Second international workshop on computer-aided software engineering. More academics and less hype than most CASE conferences, for better or worse. Sponsored by several academic institutions. Call Pamela Meyer, Index Technology (organizers), (617) 494-8200, x454.
- July 24-27** **IEEE conference on neural nets** - San Diego. The second, because the first was so successful. Contact: Richard Rea (exhibits), (619) 222-7477, Sue Varga, (619) 281-8991, or Nomi Feldman (papers), (619) 453-6222.

-
- August 1-5 **SIGGRAPH** - Atlanta. Sponsored by IEEE, ACM and SIGGRAPH.
Contact: Adele Newton, (519) 888-4534, or (312) 644-6610.
- August 2-4 **UniForum DC** - Washington, DC. Sponsored by /usr/group.
Contact: (312) 299-3131 or (800) 323-5155.
- August 15-19 **International conference on parallel processing** - DuPage
County, IL. Sponsored by Pennsylvania State University.
Contact: Roger Anderson, (415) 422-8572.
- August 15-19 **Fifth international logic programming conference** - Seattle.
Sponsored by the Association for Logic Programming. Con-
tact: Diana Nielson, (206) 543-3200.
- August 22-26 **AAAI-88** - St. Paul, MN. The seventh annual. Sponsored by
the American Association for Artificial Intelligence.
Contact: Claudia Mazzetti, (415) 328-3123.
- September 6-10 **International neural network society** - Boston. First an-
nual meeting. With tutorials and speeches by Carver Mead,
Bernard Widrow, Stephen Grossberg, Terrence Sejnowski.
Contact: J. R. Schuman Associates, (617) 237-7931.
- September 7-10 **SPA fourth annual conference** - Washington, DC. Contact:
Jayne White, (202) 452-1600.
- September 14-17 **Seybold desktop publishing conference** - Santa Clara. The
industry standard, moderated by Jonathan Seybold. Contact:
Kevin Howard, (213) 457-5850.
- September 25-27 **Agenda '89** - Carlsbad, CA. Second annual, with all-new
speakers this year. Run by Stewart Alsop, managed by
Marketing Partners, sponsored by PCW Communications. Call
Elizabeth Withrow, (415) 363-8080.
- September 25-30 **OOPSLA** - San Diego. Object-oriented Programming: systems,
languages and applications. Sponsored by ACM. Contact:
Allen Otis, Servio Logic, (503) 644-4242 or Barbara Nopar-
stak, Digitalk, (213) 645-1082. (The conference section of
OOPSLA is Wednesday through Friday (28-30), so you can
catch most of CSCW first if you miss the OOPSLA tutorials.)
- September 26-28 **Second conference on computer-supported cooperative work** -
Portland, OR. Sponsored by ACM. Contact: Suzanne Sylvia,
(617) 225-1860.
- September 26-28 **The software re-engineering symposium** - San Francisco.
Sponsored by DCI. See page 11. Contact: Marny Peabody,
(617) 475-6990.
- October 3-5 **Adam Green's dBASE^R symposium** - Burlington, MA. Three-day
conference of dBASE third parties, including clones, com-
pilers and complements. Contact: Marny Peabody at Digital
Consulting, (617) 470-3870.

- October 10-12** **Frontiers of massively parallel computation** - Fairfax, VA. Sponsored by IEEE et al. Programming languages, neural nets, graph theory, everything connected with || computing. Contact: James R. Fischer, (301) 286-9412.
- October 11-14** **Info Show** - New York City. Contact: Frank Fazio, Cahners Exposition Group, (203) 964-0000.
- October 23-27** **American Society for Information Science** - Atlanta. the 51st annual. Contact: Bob Hurd, (202) 462-1000. With speeches by Charles Marshall, AT&T; Max Hopper, American Airlines; Anthony Craig, GEISCO.
- October 23-28** **Monterey Classic** - Monterey, CA. Contact: John Baumeister, (408) 987-4200.
- October 16-19** **ADAPSO MANAGEMENT CONFERENCE** - Dallas. The software and services industry's premier gathering. Contact: Sheila Wakefield, (703) 522-5055.
- October 25-28** **Educom '88** - Washington, DC. Campaign for excellence: Education, government, industry. With Eric Bloch, NSF; Alan Kay, Apple; Ross Perot, himself. Contact: Evelyn Thompson, (609) 520-3343.
- October 30-November 3** **IIA twentieth anniversary convention** - Washington, DC. Keynote speaker: Arno Penzias, Bell Labs. Contact Mike Atkin, (202) 639-8262.
- November 2-4** **Adam Green's dBASE^R symposium** - Woodland Hills, CA. See October 3. Contact: Marny Peabody, DCI, (617) 470-3870.
- November 14-16** **Design's role in customer-based business structures for the nineties** - San Francisco. With Tom Peters. Sponsored by Design Management Inst. Call Earl Powell, (617) 236-4165.
- November 14-16** **The software re-engineering symposium** - New York City. Sponsored by DCI. See page 11. Contact: Marny Peabody, (617) 475-6990.
- November 14-18** **Comdex** - Las Vegas. The one and only, sponsored by the Interface Group. Contact: Jane Wemyss, (617) 449-6600.

1989

- February 12-15** **EDventure Holdings Personal Computing Forum** - Palm Springs, CA. Contact: Lisa Braden, (212) 758-3434.
- February 27-March 2** **UniForum** - San Francisco. Sponsored by /usr/group. Contact: Ed Palmer, (408) 986-8840.
- March 13-18** **Seybold Seminars '89** - San Francisco. Contact: Kevin Howard, (213) 457-5850.

Please let us know of any other events we should include. -- Anita Fowler

SUBSCRIPTION FORM

Please enter my subscription to **Release 1.0** at the rate of \$395 per year in the U.S. and Canada. Overseas subscriptions are \$475, airmail postage included. Payment must be enclosed. Multiple-copy rates on request. Satisfaction guaranteed or your money back.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone _____

How did you hear about Release 1.0? _____

88-5

Please fill in the information above
and send with your check payable to:

EDventure Holdings Inc.
375 Park Avenue, Suite 2503
New York, NY 10152

If you have any questions, please call us at (212) 758-3434.

Lisa Braden
Associate Publisher